Large Scale Max-Margin Multi-Label Classification with Priors

Bharath Hariharan CS1060156@CSE.IITD.ERNET.IN
Department of Computer Science and Engineering, Indian Institute of Technology, Delhi, India 110 016

Lihi Zelnik-Manor LIHI@EE.TECHNION.AC.IL
Department of Electrical Engineering, The Technion - Israel Institute of Technology, Haifa, Israel 32000

S. V. N. Vishwanathan VISHY@STAT.PURDUE.EDU
Department of Statistics, Purdue University, West Lafayette, IN 47907-2066

Manik Varma MANIK@MICROSOFT.COM
Microsoft Research India, Bangalore, India 560 080

Abstract

We propose a max-margin formulation for the multi-label classification problem where the goal is to tag a data point with a set of pre-specified labels. Given a set of L labels, a data point can be tagged with any of the 2^L possible subsets. The main challenge therefore lies in optimising over this exponentially large label space subject to label correlations.

Existing solutions take either of two approaches. The first assumes, *a priori*, that there are no label correlations and independently trains a classifier for each label (as is done in the 1-vs-All heuristic). This reduces the problem complexity from exponential to linear and such methods can scale to large problems. The second approach explicitly models correlations by pairwise label interactions. However, the complexity remains exponential unless one assumes that label correlations are sparse. Furthermore, the learnt correlations reflect the training set biases.

We take a middle approach that assumes labels are correlated but does not incorporate pairwise label terms in the prediction function. We show that the complexity can still be reduced from exponential to linear while modelling dense pairwise label correlations. By incorporating correlation priors we can overcome training set biases and improve prediction accuracy. We provide a principled interpretation of the 1-vs-All method and show that it arises as a special case of our formulation. We also develop efficient optimisation algorithms that can be orders of magnitude faster than the state-of-the-art.

1. Introduction

The objective in multi-label classification is to predict a *set* of relevant binary labels for a given input. A key aspect is dealing with the exponentially large power set of labels subject to label correlations. This is in contrast to multi-class classification where one has to predict just the single, most probable label.

Existing methods for multi-label classification take one of two approaches. In the first, labels are *a priori* assumed not to be correlated so that a predictor can be trained for each label independently (as is done in the popular 1-vs-All heuristic). This reduces training and prediction complexity from exponential in the number of labels to linear. In the second, label correlations are explicitly taken into account by incorporating pairwise, or potentially even higher order, label interactions. However, inference is intractable unless one assumes that labels are sparsely correlated. Exact inference might also not be possible in the presence of loops and most work has focused on hierarchical tree structured labels. Furthermore, label correlation weights are estimated from the training set alone and prior knowledge is rarely incorporated.

In this paper, we focus on the case where the labels are densely correlated and where the label correlations found during training might be very different from those found during testing. This is a common setting in image and video search where one has to frequently recognise categories for which training

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

data is unavailable. For instance, to recognise objects without requiring any labeled examples of the test categories, (Lampert et al., 2009) proposed an attribute based multi-label system which needs only prior knowledge about how attributes co-occur in the test categories. Existing multi-label methods are unsuitable in such scenarios.

We propose a max-margin multi-label (M3L) formulation where correlated, rather than independent, predictors are learnt but where pairwise label terms are not incorporated in the prediction function. We show that, for a problem with L labels and N training data points, the M3L formulation can be reduced from having $N2^L$ constraints to NL constraints. The formulation generalises existing approaches which assume independence, such as 1-vs-All, and provides a principled way of interpreting them. Furthermore, by sacrificing some modelling power as compared to explicit methods, the formulation can handle dense, loopy label correlations and incorporate prior knowledge efficiently.

We develop specialised algorithms for optimising the M3L formulation and demonstrate that they can be orders of magnitude faster than existing cutting plane methods (Tsochantaridis et al., 2005). In particular, for kernelised M3L, while a straight forward SMO based implementation would have taken time quadratic in the number of labels, our algorithm can train in linear time. By efficient kernel caching, we can sometimes even be an order of magnitude faster than 1-vs-All. Thus our code, available from (M3L), should also be very useful for learning independent 1-vs-All classifiers. For linear M3L, we demonstrate that we can train on the RCV1 dataset with 781,265 points and 103 labels in effectively six minutes and fully converge in eighteen. In terms of performance, we show that incorporating prior correlation information using the M3L formulation can substantially boost prediction accuracy over independent methods.

2. Related Work

Most multi-label approaches proposed in the literature try and reduce the problem to a more "canonical" one such as regression, multi-class and binary classification and ranking. We review such independent methods as well as those that take label correlations into account.

In regression methods (Ji et al., 2008; Hsu et al., 2009; Tsoumakas & Katakis, 2007), the label space is mapped onto a vector space (which might sometimes be a shared subspace of the feature space) where regression techniques can be applied straight forwardly. The primary advantage of such methods is that they can be extremely efficient if the mapped label space has significantly lower dimensionality than the original label space (Hsu et al., 2009). The disadvantage of such approaches is that the choice of an appropriate mapping might be unclear. As a result, minimising regression loss functions, such as square loss, in this space might be very efficient but might not be strongly correlated with minimising the desired multi-label loss. Furthermore, classification involves inverting the map which might not be straight forward, result in multiple solutions and might involve heuristics.

A multi-label problem with L labels can be viewed as a classification problem with 2^L classes (McCallum, 1999; Boutell et al., 2004) and standard multi-class techniques can be brought to bear. Such an approach was shown to give the best empirical results in the survey by (Tsoumakas & Katakis, 2007). Apart from computational costs, one of the main drawbacks of this approach is that most classes will have no positive training data and these label combinations can not be recognised at test time. Furthermore, the multi-class 0/1 loss is a poor approximation to the desired multi-label loss. For instance, the 0/1 loss would charge the same penalty for getting all but one of the labels right as it would for getting none of the labels right.

Binary classification can be leveraged by replicating the feature vector for each data point L times. For copy number l, an extra dimension is added to the feature vector with value l and the training label is +1 if label l is present in the label set of the original point and -1 otherwise. Due to the data replication, applying a binary classifier naively would be computationally costly and would require that complex decision boundaries be learnt. However, (Schapire & Singer, 2000) show that the problem can be solved efficiently using Boosting. A somewhat related technique is 1-vs-All (Rifkin & Khautau, 2004) which independently learns a binary classifier for each label. As we'll show in Section 3, our formulation generalises 1-vs-All to handle label correlations.

A ranking based solution was proposed in (Elisseeff & Weston, 2001). The objective was to ensure that, for every data point, all the relevant labels were ranked higher than any of the irrelevant ones. Determining the number of labels to predict for a novel point can be problematic. Some approaches address the issue by training an independent regressor while others introduce a dummy label. Posing the problem as ranking also induces a quadratic number of constraints per example which leads to a harder optimisation. This is ameliorated in (Crammer & Singer, 2003) who reduced the space complexity to linear and time complexity to sub-quadratic.

Most of the approaches mentioned above do not explicitly model label correlations ((McCallum, 1999) has a generative model which can, in principle, handle correlations but greedy heuristics are used to search over the exponential label space). In terms of discriminative methods, most work has focused on hierarchical tree, or forest, structured labels. Methods such as (Cai & Hofmann, 2007; Cesa-Bianchi et al., 2006) optimise a hierarchical loss over the tree structure but do not incorporate pairwise, or higher order, label interaction terms. For instance, (Cai & Hofmann, 2007) classify at only the leaf nodes by leveraging the ranking method of (Elisseeff & Weston, 2001). The M3N formulation of (Taskar et al., 2003) was the first to suggest max-margin learning of label interactions. Learning is exact and efficient for trees and approximate, in general, for loopy graph structures. However, learning is intractable for densely correlated labels. While the M3N formulation dealt with the Hamming loss, a more suitable hierarchical loss was introduced and efficiently optimised in (Rousu et al., 2006).

Finally, (Tsochantaridis et al., 2005) propose an iterative, cutting plane algorithm for learning in general structured output spaces. The algorithm adds the worst violating constraint to the active set in each iteration and is proved to take a maximum number of iterations independent of the size of the output space. While this algorithm can be used to learn pairwise label interactions it too can't handle a fully connected graph as the worst violating constraint can not be generally found in polynomial time. However, it can be used to learn our proposed M3L formulation but is an order of magnitude slower than the specialised optimisation algorithms we develop.

3. M3L: The Max-Margin Multi-Label Classification Primal Formulation

The objective in multi-label classification is to learn a function f which can be used to assign a *set* of labels to a point $\mathbf{x}$. We assume that N training data points have been provided of the form $(\mathbf{x}_i, \mathbf{y}_i) \in \mathbb{R}^D \times \{\pm 1\}^L$ with y_{il} being $+1$ if label l has been assigned to point i and -1 otherwise.

A principled way of formulating the problem would be to take the loss function Δ that one truly cares about and minimise it over the training set subject to regularisation or prior knowledge. Of course, since direct minimisation of most discrete loss functions is hard, we might end up minimising an upper bound on the loss, such as the hinge. The learning problem can then be formulated as the following primal

$$P_1 = \min_f \ \tfrac{1}{2}\|f\|^2 + C\sum_{i=1}^{N} \xi_i \tag{1}$$

$$\text{s. t. } f(\mathbf{x}_i, \mathbf{y}_i) \geq f(\mathbf{x}_i, \mathbf{y}) + \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i$$

$$\forall i, \mathbf{y} \in \{\pm 1\}^L \setminus \{\mathbf{y}_i\} \tag{2}$$

$$\xi_i \geq 0 \ \ \forall i \tag{3}$$

with a new point $\mathbf{x}$ being assigned labels according to $\mathbf{y}^* = \operatorname{argmax}_{\mathbf{y}} f(\mathbf{x}, \mathbf{y})$. The drawback of such a formulation is that there are $N2^L$ constraints which make direct optimisation very slow. Furthermore, classification of novel points requires 2^L function evaluations (one for each possible value of $\mathbf{y}$), which can be prohibitive at run time. In this section, we demonstrate that, under general assumptions of linearity, (P_1) can be reformulated as the minimisation of L densely correlated sub-problems each having only N constraints. At the same time, prediction cost is reduced to a single function evaluation (with complexity linear in L).

We start by making the standard assumption that $f(\mathbf{x}, \mathbf{y}) = \mathbf{w}^t(\boldsymbol{\phi}(\mathbf{x}) \otimes \boldsymbol{\psi}(\mathbf{y}))$ where $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ are the feature and label space mappings respectively, $\otimes$ is the Kronecker product and $\mathbf{w}^t$ denotes $\mathbf{w}$ transpose. To incorporate prior knowledge and correlate classifiers efficiently, we assume that labels have at most linear, possibly dense correlation so that it is sufficient to choose $\boldsymbol{\psi}(\mathbf{y}) = \mathbf{P}\mathbf{y}$ where $\mathbf{P}$ is an invertible matrix encoding all our prior knowledge about the labels.

To reduce the number of constraints from exponential to linear, we make another standard assumption of restricting ourselves to modelling loss functions that decompose over the individual labels (Taskar et al., 2003). Hence, we require that $\Delta(\mathbf{y}_i, \mathbf{y}) = \sum_{l=1}^{L} \Delta_l(\mathbf{y}_i, y_l)$ where $y_l \in \{\pm 1\}$ corresponds to label l in the set of labels represented by $\mathbf{y}$. For instance, the popular Hamming loss, amongst others, satisfies this condition. The Hamming loss $\Delta(\mathbf{y}_i, \mathbf{y})$, between a ground truth label $\mathbf{y}_i$ and a prediction $\mathbf{y}$ is given by $\Delta(\mathbf{y}_i, \mathbf{y}) = \mathbf{y}_i^t(\mathbf{y}_i - \mathbf{y})$ which is a count of twice the total number of individual labels mispredicted in $\mathbf{y}$. Note that the Hamming loss can be decomposed over the labels as $\Delta(\mathbf{y}_i, \mathbf{y}) = \sum_l 1 - y_l y_{il}$. Of course, for Δ to represent a sensible loss we also require that $\Delta(\mathbf{y}_i, \mathbf{y}) \geq \Delta(\mathbf{y}_i, \mathbf{y}_i) = 0$.

Under these assumptions, (P_1) can be expressed as

$$P_1 \equiv \min_{\mathbf{w}} \tfrac{1}{2}\mathbf{w}^t\mathbf{w} + C\sum_{i=1}^{N} \max_{\mathbf{y} \in \{\pm 1\}^L} [\Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w}^t\boldsymbol{\phi}(\mathbf{x}_i) \otimes \mathbf{P}(\mathbf{y} - \mathbf{y}_i)] \tag{4}$$

where the constraints have been moved into the objective and $\xi_i \geq 0$ eliminated by including $\mathbf{y} = \mathbf{y}_i$ in the maximisation. To simplify notation, we express the vector $\mathbf{w}$ as a $D \times L$ matrix $\mathbf{W}$ so that

$$P_1 \equiv \min_{\mathbf{W}} \tfrac{1}{2}\,\mathrm{Trace}(\mathbf{W}^t\mathbf{W}) + C\sum_i \max_{\mathbf{y}}[\Delta(\mathbf{y}_i,\mathbf{y}) + (\mathbf{y}-\mathbf{y}_i)^t\mathbf{P}^t\mathbf{W}^t\boldsymbol{\phi}(\mathbf{x}_i)] \quad (5)$$

Substituting $\mathbf{Z} = \mathbf{WP}$, $\mathbf{R} = \mathbf{P}^t\mathbf{P} \succ 0$ and using the identity $\mathrm{Trace}(\mathbf{ABC}) = \mathrm{Trace}(\mathbf{CAB})$ results in

$$P_1 \equiv \min_{\mathbf{Z}} \tfrac{1}{2}\sum_{l=1}^{L}\sum_{k=1}^{L} R^{-1}_{lk}\mathbf{z}_l^t\mathbf{z}_k + C\sum_i \max_{\mathbf{y}} \left[\sum_{l=1}^{L}[\Delta_l(\mathbf{y}_i,y_l) + (y_l - y_{il})\mathbf{z}_l^t\boldsymbol{\phi}(\mathbf{x}_i)]\right] \quad (6)$$

where $\mathbf{z}_l$ is the l^{th} column of $\mathbf{Z}$. Note that the terms inside the maximisation break up independently over the L components of $\mathbf{y}$. It is therefore possible to interchange the maximisation and summation to get

$$P_1 \equiv \min_{\mathbf{Z}} \tfrac{1}{2}\sum_{l=1}^{L}\sum_{k=1}^{L} R^{-1}_{lk}\mathbf{z}_l^t\mathbf{z}_k + C\sum_i\sum_{l=1}^{L} \left[\max_{y_l\in\{\pm1\}}[\Delta_l(\mathbf{y}_i,y_l) + (y_l - y_{il})\mathbf{z}_l^t\boldsymbol{\phi}(\mathbf{x}_i)]\right] \quad (7)$$

This leads to an equivalent primal formulation (P_2) as the summation of L correlated problems, each having N constraints which is significantly easier to optimise.

$$P_2 = \sum_{l=1}^{L} S_l \quad (8)$$

$$S_l = \min_{\mathbf{Z},\boldsymbol{\xi}} \tfrac{1}{2}\mathbf{z}_l^t\sum_{k=1}^{L} R^{-1}_{lk}\mathbf{z}_k + C\sum_{i=1}^{N}\xi_{il} \quad (9)$$

$$\text{s. t. } 2y_{il}\mathbf{z}_l^t\boldsymbol{\phi}(\mathbf{x}_i) \geq \Delta_l(\mathbf{y}_i,-y_{il}) - \xi_{il} \quad (10)$$

$$\xi_{il} \geq \Delta_l(\mathbf{y}_i,y_{il}) \quad (11)$$

Furthermore, a novel point $\mathbf{x}$ can be assigned the set of labels for which the entries of $\mathrm{sign}(\mathbf{Z}^t\boldsymbol{\phi}(\mathbf{x}))$ are $+1$. This corresponds to a single function evaluation as compared to the 2^L in the original case.

Note that the L classifiers in $\mathbf{Z}$ are not independent but correlated by $\mathbf{R}$ – a positive definite matrix encoding our prior knowledge about label correlations. Depending on the application, $\mathbf{R}$ can be dense and even have negative entries. Due to this, the number of constraints would have remained exponential had we made f quadratic in $\mathbf{y}$ by explicitly including pairwise terms as in (Taskar et al., 2003). Also, note that we deliberately chose not to include bias terms $\mathbf{b}$ in f even though the reduction from (P_1) to (P_2) would still have gone through and the resulting kernelised optimisation been more or less the same (see Section 6.1). However, we would then have had to regularise $\mathbf{b}$ and correlate it using $\mathbf{R}$. Otherwise $\mathbf{b}$ would have been a free parameter capable of undoing the effects of $\mathbf{R}$ on $\mathbf{Z}$. Therefore, rather than explicitly have $\mathbf{b}$ and regularise it, we implicitly simulate $\mathbf{b}$ by adding an extra dimension to the feature vector. This has the same effect while keeping optimisation simple.

Equivalence with 1-vs-All If label correlation information is not included, i. e. $\mathbf{R} = \mathbf{I}$, then (P_2) decouples into L completely independent sub-problems each of which can be tackled in isolation. In particular, for the Hamming loss we get

$$S_l = \min_{\mathbf{z}_l,d_l,\boldsymbol{\xi}} \tfrac{1}{2}\mathbf{z}_l^t\mathbf{z}_l + 2C\sum_{i=1}^{N}\xi_i \quad (12)$$

$$\text{s. t. } y_{il}\mathbf{z}_l^t\boldsymbol{\phi}(\mathbf{x}_i) \geq 1 - \xi_i \quad (13)$$

$$\xi_i \geq 0 \quad (14)$$

Thus, S_l reduces to an independent binary classification sub-problem where the positive class contains all training points tagged with label l and the negative class containing all other points. This is exactly the strategy used in the popular 1-vs-All heuristic and we can therefore now make explicit the assumptions underlying this technique. The only difference is that one should charge a misclassification penalty of $2C$ to be consistent with the original primal formulation.

4. The M3L Dual Formulation

The dual of (P_2) has similar properties in that it can be viewed as the maximisation of L related problems which decouple into independent binary SVM classification problems when $\mathbf{R} = \mathbf{I}$. The dual is easily derived if we rewrite (P_2) in vector notation. Defining $\mathbf{Y}_l = \mathrm{diag}([y_{1l},\ldots,y_{Nl}])$, $\mathbf{K}_{\mathbf{x}} = \boldsymbol{\phi}^t(\mathbf{X})\boldsymbol{\phi}(\mathbf{X})$ and $\boldsymbol{\Delta}_l^{\pm} = [\Delta_l(\mathbf{y}_1,\pm y_{1l}),\ldots,\Delta_l(\mathbf{y}_N,\pm y_{Nl})]^t$ we get the following Lagrangian

$$L = \sum_{l=1}^{L}(\tfrac{1}{2}\sum_{k=1}^{L} R^{-1}_{lk}\mathbf{z}_l^t\mathbf{z}_k + C\mathbf{1}^t\boldsymbol{\xi}_l - \boldsymbol{\beta}_l^t(\boldsymbol{\xi}_l - \boldsymbol{\Delta}_l^+) - \boldsymbol{\alpha}_l^t(2\mathbf{Y}_l\boldsymbol{\phi}^t(\mathbf{X})\mathbf{z}_l - \boldsymbol{\Delta}_l^- + \boldsymbol{\xi}_l)) \quad (15)$$

with the optimality conditions being

$$\boldsymbol{\nabla}_{\mathbf{z}_l}L = 0 \Rightarrow \sum_{k=1}^{L} R^{-1}_{lk}\mathbf{z}_k = 2\boldsymbol{\phi}(\mathbf{X})\mathbf{Y}_l\boldsymbol{\alpha}_l \quad (16)$$

$$\boldsymbol{\nabla}_{\boldsymbol{\xi}_l}L = 0 \Rightarrow C\mathbf{1} - \boldsymbol{\alpha}_l - \boldsymbol{\beta}_l = \mathbf{0} \quad (17)$$

Substituting these leads to the following dual

$$D_2 = \max_{\mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1}} \sum_{l=1}^{L} \boldsymbol{\alpha}_l^t (\boldsymbol{\Delta}_l^- - \boldsymbol{\Delta}_l^+) - 2 \sum_{l=1}^{L} \sum_{k=1}^{L} R_{lk} \boldsymbol{\alpha}_l^t \mathbf{Y}_l \mathbf{K} \mathbf{Y}_k \boldsymbol{\alpha}_k \qquad (18)$$

5. Optimisation

The M3L dual is similar to the standard SVM dual. Existing optimisation techniques can therefore be brought to bear. However, the dense structure of $\mathbf{R}$ couples all NL dual variables and simply porting existing solutions leads to very inefficient code. We show that, with book keeping, we can easily go from an $O(L^2)$ algorithm to an $O(L)$ algorithm. Furthermore, by re-utilising the kernel cache, our algorithms can be very efficient even for non-linear problems. We treat the kernelised and linear M3L cases separately.

5.1. Kernelised M3L

We optimise the M3L dual using coordinate ascent with second order variable selection. Two dual variables are chosen for optimisation at every iteration. The first variable is chosen to be the one with the maximum projected gradient magnitude. The second is chosen so as to maximise second order dual progress. We maintain gradients $\boldsymbol{\nabla}_{\boldsymbol{\alpha}} D_2$ (where D_2 is now overloaded to mean the dual objective) in order to pick variables efficiently. After optimisation the dual variable α_{il}, all gradients need to be updated as

$$\boldsymbol{\nabla}^{new}_{\alpha_{jk}} D_2 = \boldsymbol{\nabla}^{old}_{\alpha_{jk}} D_2 - 4 y_{il} y_{jk} R_{kl} K_{ij} (\alpha_{il}^{new} - \alpha_{il}^{old})$$

For dense $\mathbf{R}$ this implies an inefficient $O(L^2)$ algorithm as NL gradients need to be updated in each iteration. To mitigate this problem, we optimise along a single label for L consecutive iterations maintaining gradients for only that label. Now switching to the label with the maximal dual progress might seem difficult as gradients for the other labels haven't been maintained. However, by simple book keeping, all the other gradients can be updated in $O(NL)$ time as

$$\boldsymbol{\nabla}^{new}_{\alpha_{jk}} D_2 = \boldsymbol{\nabla}^{old}_{\alpha_{jk}} D_2 - 4 y_{jk} R_{kl} u_{jl} \qquad (19)$$

where $u_{jl} = \sum_{i=1}^{N} y_{il} K_{ij} (\alpha_{il}^{new} - \alpha_{il}^{old})$ and l indexes the label along which we have been optimising. This leads to an efficient $O(L)$ algorithm with dramatically reduced runtime even though slightly more iterations were needed for convergence (see (M3L) for a proof). We also employed an effective kernel cache.

5.2. Linear M3L

We build on top of the dual coordinate ascent algorithm of (Hsieh et al., 2008). While a set of active points is still maintained we no longer maintain gradients or a cache. During each pass over the active set, dual variables are randomly picked and optimised analytically. Points at bound having gradient magnitudes outside the range of currently maintained extremal gradients are discarded. Extremal gradients are re-estimated at the end of each pass and if they are too close to each other the active set is expanded to include all training points.

A straight forward implementation with globally maintained extremal gradients again leads to slow training. Essentially, if the classifier for a particular label has not yet converged, then it can force a large active set even though most points would not be considered by the other classifiers. We therefore implemented separate active sets for each label but coupled the maintained extremal gradients via $\mathbf{R}$. This was empirically found to decrease training time.

6. Experiments

In this section we first compare the performance of our optimisation algorithms and then evaluate how prediction accuracy can be improved by incorporating prior knowledge about label correlations.

6.1. Optimisation Experiments

The cutting plane algorithm in SVMStruct (Tsochantaridis et al., 2005) is an excellent general purpose algorithm that can be used to optimise the original M3L formulation (P_1). In each iteration, the approximately worst violating constraint is added to the active set and the algorithm is proved to take a maximum number of iterations independent of the size of the output space. The algorithm has a user defined parameter ϵ for the amount of error that can be tolerated in finding the worst violating constraint.

We compared the SVMStruct algorithm to our M3L implementation on an Intel Xeon 2.67 GHz machine with 8GB RAM. Even on medium scale problems with linear kernels, our M3L implementation was nearly a hundred times faster than SVMStruct. For example, on the Media Mill dataset (Snoek et al., 2006) with 101 labels and ten, fifteen and twenty thousand training points, our M3L code took 19, 37 and 55 seconds while SVMStruct took 1995, 2998 and 7198 seconds respectively. On other datasets SVMStruct ran out of RAM or failed to converge in a reasonable amount of time (even after tuning ϵ). This demonstrates that ex-

Table 1. Timing results for our linear M3L (LM3L) and kernelised M3L (KM3L) optimisation algorithms on datasets with N training points, D features and L labels. See text for details.

(a) Animals with Attributes: D=252, L=85.

N	Linear Kernel (s)				RBF Kernel (s)	
	1-vs-All LibLinear	LM3L	1-vs-All LibSVM	KM3L	1-vs-All LibSVM	KM3L
2,000	3	7	234	15	250	20
10,000	48	51	5438	245	6208	501
15,000	68	74	11990	500	13875	922
24,292	102	104	29328	1087	34770	3016

(b) RCV1: D=47,236(sparse), L=103.

N	Linear Kernel (s)				RBF Kernel (s)	
	1-vs-All LibLinear	LM3L	1-vs-All LibSVM	KM3L	1-vs-All LibSVM	KM3L
2,000	7	4	54	6	139	11
10,000	23	27	743	110	1589	177
15,000	33	43	1407	230	2893	369
23,149	45	57	2839	513	5600	817

(c) Siam: D=30,438(sparse), L=22.

N	Linear Kernel (s)				RBF Kernel (s)	
	1-vs-All LibLinear	LM3L	1-vs-All LibSVM	KM3L	1-vs-All LibSVM	KM3L
2,000	1	1	27	5	43	7
10,000	2	2	527	126	775	185
15,000	3	3	1118	288	1610	422
21,519	5	5	2191	598	3095	878

(d) Media Mill: D=120, L=101.

N	Linear Kernel (s)				RBF Kernel (s)	
	1-vs-All LibLinear	LM3L	1-vs-All LibSVM	KM3L	1-vs-All LibSVM	KM3L
2,000	2	2	11	2	15	6
10,000	18	19	456	57	505	123
15,000	35	37	1014	124	1107	275
25,000	62	75	2662	337	2902	761
30,993	84	97	4168	527	4484	1162

plicitly reducing the number of constraints from exponential to linear and implementing a specialised solver can lead to a dramatic reduction in training time.

As the next best thing, we benchmark our performance against the 1-vs-All method, even though it can't learn correlated classifiers. In the linear case, we compare to 1-vs-All trained by running LibLinear (Fan et al., 2008) and LibSVM (Chang & Lin, 2001) independently over each label. For non-linear kernels we compare to 1-vs-All trained using LibSVM. In each case, we set $\mathbf{R} = \mathbf{I}$, so that M3L reaches exactly the same solution as LibSVM and LibLinear. Also, we avoided repeated disk I/O by reading the data into RAM and using LibLinear and LibSVM's API's.

Table 1 lists the variation in training time with the number of training examples on the Animals with Attributes (Lampert et al., 2009), Media Mill (Snoek et al., 2006), Siam (SIA) and RCV1 (Lewis et al., 2004) datasets. The training times of linear M3L (LM3L) and LibLinear are comparable, with LibLinear being slightly faster. The training time of kernelised M3L (KM3L) are significantly lower than LibSVM, with KM3L sometimes being as much as 30 times faster. This is because KM3L can efficiently leverage the kernel cache across all labels while LibSVM has to build the cache from scratch each time. This isn't an issue in linear M3L and LibLinear as no kernel cache is maintained. Thus, even though M3L generalises 1-vs-All, its training time can be comparable, and sometimes, even significantly lower.

Finally, to demonstrate that our code scales to large problems, we train linear M3L on RCV1 with 781,265 points and 103 labels. Table 2 charts dual progress

Table 2. Linear M3L training on RCV1.

Time(s)	Dual	Train Error(%)	Test Error(%)
60	1197842	0.86	0.98
183	1473565	0.74	0.84
300	1492664	0.72	0.83
338	1494012	0.72	0.82
345	1494050	0.72	0.82
353	1494057	0.72	0.82
1080	1494057	0.72	0.82

and train and test error with time. As can be seen, the model is nearly fully trained in under six minutes and converges in eighteen.

6.2. Incorporating Prior Knowledge

An interesting scenario, which has only recently been introduced in computer vision, is of recognising object categories that have never been seen during training but about whom prior information might be available (Lampert et al., 2009). If the training and test categories share a common set of object attributes, and the attributes for each test category are known *a priori*, then (Lampert et al., 2009) show how a multi-label system can be used to predict significantly better than chance. We investigate whether basic attribute prediction can be improved if the distribution of test categories is also known *a priori*.

The Animals with Attributes dataset (Lampert et al., 2009) has 40 training animal categories and 10 disjoint test animal categories which share a common set of 85 attributes. The attributes are densely correlated and form a fully connected graph. Each image in the database contains a dominant animal and is labelled with its 85 attributes. There are 24,292 training images and 6,180 test images. We use 252 dimensional PHOG features that are provided by the authors. Training times are reported in Table (1a).

We start by visualising the influence of $\mathbf{R}$. We randomly sample 200 points from the training set and discard all but two of the attributes – "black" and "weak". These two attributes were selected as they are very weakly correlated on our training set, with a correlation coefficient of 0.2, but have a strong negative correlation of -0.76 on the test animals (Leopards, Giant Pandas, Humpback Whales, *etc*). Figure 1 plots the Hamming loss on the test set as we set $\mathbf{R} = [1\ r; r\ 1]$, plug it into the M3L formulation, and vary r from -1 to +1. Learning independent classifiers for the two attributes ($r = 0$) can lead to a Hamming loss of 25% because of the mismatch between training and test sets. This can be made even worse by choosing an incorrect prior that forces the two classifiers to be correlated ($r = +1$). However, if our priors are generally correct, then negatively correlating the classifiers lowers prediction error.

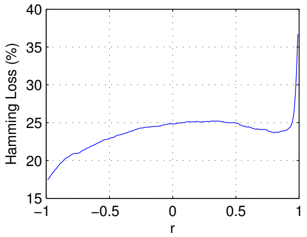

Figure 1. Test Hamming loss versus classifier correlation.

We now evaluate performance quantitatively on the same training set but with all 85 labels. For the M3L formulation we set $\mathbf{R} = \sum_{c=1}^{10} p(c)\mathbf{y}_c\mathbf{y}_c^t$ where $\mathbf{y}_c$ is the known attribute vector for test category c and $p(c)$ is the probability of occurrence of class c during testing (which we require as additional prior information). Under this setup, learning independent classifiers using 1-vs-All yields a Hamming loss of 29.38%. The Hamming loss for M3L, with the specific choice of $\mathbf{R}$, is 26.35%. This decrease in error is very significant given that 1-vs-All, trained on all 24,292 training points, only manages to reduce error to 28.64%. Thus M3L, with extra knowledge, in the form of just test category distributions, can dramatically reduce test error. The results also compare favourably to other independent methods such as BoostTexter (Schapire & Singer, 2000) (30.28%), power set multi-class classification (32.70%), 5 nearest neighbours (31.79%), regression (Hsu et al., 2009) (29.38%) and ranking (Crammer & Singer, 2003) (34.84%).

Benchmark Datasets Our interest is in multi-label problems where the training set statistics do not reflect the test set statistics. Unfortunately, most benchmark datasets do not have this property. We therefore take the Siam, Media Mill and RCV1 datasets and create train and test splits where the labels are correlated dif-

Table 3. Test Hamming loss (%) on benchmark datasets.

Method	Siam	Media Mill	RCV1
M3L	**8.41**	**3.78**	**3.45**
1-vs-All	11.15	4.69	4.25
BoostTexter	12.91	4.91	4.12
Power Set	14.01	6.27	3.71
Regression	11.19	4.69	4.26
Ranking	9.41	9.06	5.67
5-NN	12,51	4.74	4.47

ferently. The **R** matrix, encoding label correlation information, is estimated from a disjoint third set whose points are not used for training. Table 3 compares the performance of various methods. In these scenarios, it would appear that M3L can consistently leverage prior knowledge to outperform independent methods.

7. Conclusions

We developed the M3L formulation for learning max-margin multi-label classification with prior knowledge about densely correlated labels. We showed that the number of constraints could be reduced from exponential to linear and, in the process, generalised 1-vs-All classification. We also developed efficient optimisation algorithms that were orders of magnitude faster than the standard cutting plane method. Our kernelised algorithm was significantly faster than even 1-vs-All and hence our code, available from (M3L), can also be used for efficient independent learning. Finally, we demonstrated that incorporating prior knowledge using M3L could improve prediction accuracy over independent methods and that M3L trained on 200 points could outperform 1-vs-All trained on nearly 25,000.

Acknowledgements

We would like to thank Alekh Agarwal, Brendan Frey and Sunita Sarawagi for helpful discussions.

References

M3L code and convergence proof `http://research.microsoft.com/~manik/code/M3L/download.html`.

The SIAM Text Mining Competition 2007 http://www.cs.utk.edu/tmw07/).

Boutell, M., Luo, J., Shen, X., and Brown, C. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

Cai, L. and Hofmann, T. Exploiting known taxonomies in learning overlapping concepts. In *IJCAI*, pp. 714–719, 2007.

Cesa-Bianchi, N., Gentile, C., and Zaniboni, L. Incremental algorithms for hierarchical classification. *JMLR*, 7:31–54, 2006.

Chang, C.-C. and Lin, C.-J. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Crammer, K. and Singer, Y. A family of additive online algorithms for category ranking. *JMLR*, 3:1025–1058, 2003.

Elisseeff, A. and Weston, J. A kernel method for multilabelled classification. In *NIPS*, pp. 681–687, 2001.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. Liblinear: A library for large linear classification. *JMLR*, 9:1871–1874, 2008.

Hsieh, C.-J., Chang, K.-W., Lin, C.-J., Keerthi, S. S., and Sundarajan, S. A dual coordinate descent method for large-scale linear svm. In *ICML*, 2008.

Hsu, D., Kakade, S., Langford, J., and Zhang, T. Multi-label prediction via compressed sensing. In *NIPS*, 2009.

Ji, S., Sun, L., Jin, R., and Ye, J. Multi-label multiple kernel learning. In *NIPS*, pp. 777–784, 2008.

Lampert, C. H., Nickisch, H., and Harmeling, S. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.

Lewis, D., Yang, Y., Rose, T., and Li, F. RCV1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, 2004.

McCallum, A. Multi-label text classification with a mixture model trained by EM. In *AAAI 99 Workshop on Text Learning*, 1999.

Rifkin, R. and Khautau, A. In defense of one-vs-all classification. *JMLR*, 5:101–141, 2004.

Rousu, J., Saunders, C., Szedmak, S., and Shawe-Taylor, J. Kernel-based learning of hierarchical multilabel classification models. *JMLR*, 7:1601–1626, 2006.

Schapire, R. E. and Singer, Y. Boostexter: A boosting-based system for text categorization. *ML*, 39(2/3): 135–168, 2000.

Snoek, C., Worring, M., van Gemert, J., Geusebroek, J.-M., and Smeulders, A. The challenge problem for automated detection of 101 semantic concepts in multimedia. In *ACM Multimedia*, pp. 421–430, 2006.

Taskar, B., Guestrin, C., and Koller, D. Max-margin markov networks. In *NIPS*, 2003.

Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

Tsoumakas, G. and Katakis, I. Multi-label classification: An overview. *Int. Journal of Data Warehousing and Mining*, 3(3):1–13, 2007.